基于改进鱼群算法的几何约束求解 *

高雪瑶，刘云腾

(哈尔滨理工大学 计算机科学与技术学院，哈尔滨 150080)

摘　要：几何约束求解是 CAD 建模中的关键技术。针对求解质量不高和求解速度慢的问题，进行了研究。提出了一种鱼群算法和混沌算法相结合的几何约束求解方法。首先，将 CAD 模型中的几何约束关系表示为一组代数方程组；然后，利用代数方程组来构造目标函数。将几何约束求解问题转换为目标函数的优化问题。最后，使用混沌算法来改进鱼群算法以寻找目标函数的最优解。实验结果表明：该方法可以有效地解决几何约束问题。

关键词：几何约束求解；鱼群算法；混沌算法；目标函数

中图分类号：TP391.7　　doi: 10.3969/j.issn.1001-3695.2017.10.0768

Geometric constraint solving based on improved fish swarm algorithm

Gao Xueyao, Liu Yunteng

(*School of Computer Science & Technology*, *Harbin University of Science & Technology*, *Harbin* 150080, *China*)

Abstract: Geometric constraint solving is a key technique in CAD modeling. This paper researched the problem that solving quality was not high and solving speed was slow. It proposed a method of geometric constraint solving, which combined fish swarm algorithm and chaos algorithm. Firstly, this paper denoted geometric constraint relations in CAD model as a group of algebraic equations. Secondly, it used this group of algebraic equations to construct objective function. This paper transformed a problem of geometric constraint solving into a problem of objective function optimization. Finally, it used chaos algorithm to improve fish swarm algorithm in order to find an optimal solution of objective function. Experimental results show that this method can effectively solve geometric constraint problems.

Key Words: geometric constraint solving; fish swarm algorithm; chaos algorithm; objective function

0　引言

在计算机辅助设计中，几何约束求解是一个关键性问题。目前，国内外学者针对这一问题开展了大量的研究工作。曹春红使用了量子行为粒子群算法来解决几何约束问题。在该算法中，利用波函数来确定粒子的状态[1]。戴春来通过添加部分约束来减少复杂耦合约束求解过程中所需要的变量数目。同时，他提出了一种基于自由度分析的耦合几何约束求解方法[2]。周友行将参数化设计过程中的几何约束求解转换为代数方程组的求解，提出了一种可选指数进制变步长数值求解算法[3]。曹春红以几何约束图中节点的度为基础，提出了一种 D-Tree 分解算法，可以解决欠约束问题和完备约束问题[4]。Thierry 使用粒子弹簧系统的数值迭代法来处理刚性和非刚性的几何约束问题。在这种方法中，粒子对应于几何实体，弹簧对应于几何约束[5]。Mathis 给出了一种新的几何约束求解框架，使用距离几何来保持刚体运动的不变性，采用分解方法来处理模型中约束过多的问题[6]。Gao 提出了将二维和三维几何约束问题分解为 C-Tree 的算法。利用该算法能够把一个规模较大的几何约束问题分解为多个子问题。同时，综合这些子问题的解得到原问题的解[7]。De 给出了一种 3D 建模工具，将几何约束以多种形式反馈给用户，能够帮助用户方便地定义几何约束，创建正确的 3D 模型[8]。郭希娟利用 D-优化算法将一个欠约束问题分解成若干个单位问题，再对单位问题进行逐一求解。实现了对集群的有效重组，降低了几何约束求解的难度[9]。曹春红以雁群飞行特征为基础，保持了粒子的多样性。同时，利用粒子之间的有用信息来增强相互的竞争能力[10]。曹春红使用交互更新模式的量子遗传算法来解决几何约束问题。在该算法中，通过量子门变换来实现遗传算法中的交叉操作[11]。黄学良利用混合图来表示三维几何约束系统，根据混合图的动态更新机制，来确定约束系统动态变化时的影响域。通过这种增量求解的方法，提高了三维几何约束系统的求解效率[12]。李文辉通过把几何约束系统分解为一棵扩展 C-树，提出了一种基于增量的 LMA 算法。利用该

基金项目：国家自然科学基金资助项目（61502124，60903082）；中国博士后科学基金资助项目（2014M560249）；黑龙江省自然科学基金资助项目（F2015041，F201420）

作者简介：高雪瑶（1979-），女，黑龙江哈尔滨人，副教授，硕导，博士，主要研究方向为计算机图形学与 CAD（xueyao_gao@163.com）；刘云腾（1992-），男，硕士研究生，主要研究方向为计算机图形学与 CAD.

方法可以对几何约束系统进行最大化分解[13]。

本文使用代数方程组来表示几何约束问题。构造目标函数，将代数方程组的求解过程转换为函数的优化过程。使用混沌搜索方法来改进鱼群算法，以寻找几何约束问题的最优解。

几何约束问题的表示

几何约束是指几何元素之间应该满足的关系。根据约束属性不同，可以分为尺寸约束和结构约束。尺寸约束描述了几何元素的形状和大小。结构约束描述了几何元素之间应该满足的拓扑关系，即几何元素之间的相对位置。无论是尺寸约束还是结构约束，都可以使用参数来进行表示。因此，几何约束求解是寻找满足所有几何约束关系的模型参数最优解的过程。

几何约束问题可以使用二元组(E, C)来进行表示。其中，$E=\{e_1, e_2, ..., e_k\}$是模型中所有几何元素的集合；$C=\{c_1, c_2, ..., c_m\}$是模型中所有几何约束关系的集合。使用代数方程$f_i(x_1, x_2, ..., x_n)=0$来表示几何约束关系$c_i$(i=1, 2, ..., m)。$C$中的所有几何约束关系就形成了一个代数方程组，如式(1)所示。

$$\begin{cases} f_1(x_1, x_2, ..., x_n)=0 \\ f_2(x_1, x_2, ..., x_n)=0 \\ \quad ... \\ f_m(x_1, x_2, ..., x_n)=0 \end{cases} \tag{1}$$

其中：$f_i(x_1, x_2, ..., x_n)=0$为几何约束方程，$m$为模型所包含的几何约束关系的个数；$x_1, x_2, …, x_n$为几何元素的参数。几何约束问题的求解就转换为代数方程组(1)的求解。当m和n的数值较大时，代数方程组(1)的求解过程将是非常复杂的。因此，构造目标函数$F(X)$，如式(2)所示。

$$F(X)=F(x_1, x_2, ..., x_n)=\sum_{i=1}^{m} f_i^2(x_1, x_2, ..., x_n)=0 \tag{2}$$

式(1)和(2)具有相同的解。$F(X)$是一个大于等于0的数。当$F(X)$越小时，其X的取值就越接近于代数方程组(1)的解。因此，几何约束问题的求解将转换为求使$F(X)$取最小值时X解的过程。

本文采用混沌算法来改进鱼群算法，并对目标函数$F(X)$进行优化求解。以保证算法具有较强的全局搜索能力，并能及时跳出局部最优。

1　基于改进鱼群算法的几何约束关系求解

鱼群算法是一种群智能搜索算法，用以解决函数优化问题。在一片水域中，食物浓度越高的地方鱼的数量就会越多。在寻找食物的过程中，鱼群会出现随机行为、觅食行为、聚群行为和追尾行为。鱼群算法模仿了鱼群在水中运动的规律。

在鱼群算法中，其参数主要包括：n代表求解维数，*try_number*为最大试探次数，*step*为步长，*visual*为鱼群个体的视野范围，*delta*为拥挤度因子，N为算法迭代次数，δ为算法精度。鱼群个体的位置用$X=(x_1, x_2, ..., x_n)$来表示。其中，x_i(i=1, 2, ..., n)为寻优变量。$dis_{i,j}$表示从位置X_i到位置X_j的距离。此处，用欧式距离公式来计算$dis_{i,j}$。食物浓度使用式(2)中的$F(X)$来表示。此处，函数值越小表示食物浓度越高。

利用式(3)来修改步长*step*和视野*visual*。

$$\begin{cases} visual = visual*\alpha + visual_{\min} \\ step = step*\alpha + step_{\min} \\ \alpha = \exp(-30\times(k/N)^s) \end{cases} \tag{3}$$

其中：$visual_{\min}$表示最小视野范围，$step_{\min}$为最小步长，S为大于1的整数。

a)随机行为。鱼群个体在视野范围内，从当前位置X_i随机选择下一个位置X_{next}的行为。

b)觅食行为。设鱼群个体的当前位置为X_i，在邻域内随机选择下一个位置X_{next}。若$F(X_{next})>F(X_i)$，则鱼群个体根据式(4)进行移动。否则，继续选择其他位置进行比较。当比较次数达到*try_number*后，还没有找到食物浓度比较高的位置，执行随机行为。其中，*rand*()为随机函数。

$$X_{prey} = X_i + (X_{next} - X_i)*step*rand()/dis_{i,next} \tag{4}$$

c)聚群行为。设鱼群个体当前位置为X_i，搜索邻域内的鱼群个体总数为n_f，中心位置为X_c。若$F(X_c)/n_f>delta*F(X_i)$，则X_c处的食物浓度较高且不太拥挤。根据公式(5)进行移动，否则，执行觅食行为。

$$X_{swarm} = X_i + (X_c - X_i)*step*rand()/dis_{i,c} \tag{5}$$

d)追尾行为。设鱼群个体当前位置为X_i，搜索邻域内的鱼群个体总数为n_f。从n_f个个体中找到具有最小函数值的个体，其位置为X_{vbest}。若$F(X_{vbest})/n_f>delta*F(X_i)$，则$X_{vbest}$处的食物浓度较高且不太拥挤。根据式(6)进行移动，否则，执行觅食行为。

$$X_{follow} = X_i + (X_{vbest} - X_i)*step*rand()/dis_{i,vbest} \tag{6}$$

e)行为选择。每个鱼群个体在执行完觅食行为、聚群行为和追尾行为后，得到X_{prey}，X_{swarm}和X_{follow}。计算$F(X_{prey})$、$F(X_{swarm})$和$F(X_{follow})$的值，将具有最小函数值的位置记为X_{b_best}，作为鱼群个体要移动的位置。

f)公告板。记录鱼群个体已发现的食物浓度最高的位置，用X_{best}表示。在所有鱼群个体通过觅食行为、聚群行为和追尾行为更新完自身的位置后，找出具有最小函数值的个体，其位置为X_{cbest}。若$F(X_{cbest})<F(X_{best})$，则$X_{best}=X_{cbest}$，$F(X_{best})=F(X_{cbest})$；否则保持不变。

鱼群算法是一种群智能优化算法，具有全局寻优特性。当搜索区域过大时，算法收敛速度减慢，求解精度降低，容易陷入局部最优。

混沌是一种表面无序，但内在具有一定规律性的非线性现象。这种非线性现象使它具有随机性、遍历性和规律性。混沌现象具有类似随机变量的表现，可以在一定时间内不重复地遍历空间内的所有状态。混沌现象可以通过特定的代数方程来进行表示。混沌算法在解空间内随机搜索，从而遍历空间内的所

有可行解。本文利用了 Logistic 映射模型，如式(7)所示。

$$z_{k+1}=\mu z_k\left(1-z_k\right) \tag{7}$$

其中：z 为映射变量，k 表示迭代次数。μ 为控制变量，取值范围为 1 到 4。当 μ 的数值为 4 时，是一种完全混沌状态。此时，该映射所生成的混沌序列的概率密度函数如式(8)所示。

$$\rho(z)=\begin{cases}\dfrac{1}{\pi\sqrt{z(1-z)}} & 0<z<1\\ 0 & 其他\end{cases} \tag{8}$$

从式(8)可以看出：当 μ 的数值为 4 时，Logistic 映射模型所产生的混沌序列可以对某一邻域内的所有空间进行全局搜索。

本文使用混沌算法来改进鱼群算法，其主要思想是：在每个鱼群个体完成行为选择之后，得到了 X_{b_best}。在鱼群个体的视野范围之内，使用混沌算法对 X_{b_best} 的每个分量进行搜索，得到位置 X_{new}。X_{b_best} 的第 i 个分量 $X_{b_best}(i)$的更新过程如式(9)所示。若 $F(X_{new})<F(X_{b_best})$，则 $X_{b_best}=X_{new}$。使用混沌算法来进行改进，可以使鱼群算法跳出局部最优，提高了搜索求解效率。

$$\begin{cases}\Delta x=-visual+2*visual*z\\ X_{new}(i)=X_{b_best}(i)+\Delta x\end{cases} \tag{9}$$

改进鱼群算法的步骤如下所示：

a)使用方程 $f_i(x_1, x_2, ..., x_n)=0$ 来表示几何约束关系 c_i(i=1, 2, ..., m)。

b)根据式(2)来构造目标函数 $F(X)$。

c)初始化参数 N、δ、$step$ 和 $visual$ 的数值，当前迭代次数 k=1。

d)初始化所有鱼群个体的位置，计算其函数值。

e)从所有鱼群个体中，找出具有最小函数值的个体，将其位置放入公告板中，记为 X_{best}。

f)鱼群个体分别执行觅食行为、聚群行为和追尾行为，得到 X_{prey}、X_{swarm} 和 X_{follow}，根据函数值 $F(X_{prey})$、$F(X_{swarm})$和 $F(X_{follow})$来选择该个体的最优位置 X_{b_best}。

g)从 X_{b_best} 出发进行混沌搜索，获得 X_{new}。若 $F(X_{new})<F(X_{b_best})$，则 $X_{b_best}=X_{new}$，$F(X_{b_best})=F(X_{new})$。

h)鱼群个体移动到 X_{b_best} 处，更新当前位置。

i)鱼群中的每个个体执行步骤(6)~(8)。

j)从所有鱼群个体中，找出具有最小函数值的个体，其位置为 X_{cbest}。若 $F(X_{cbest})<F(X_{best})$，则 $X_{best}=X_{cbest}$，$F(X_{best})=F(X_{cbest})$；否则保持不变。

k)若 $k\leq N$ 或未达到算法精度 δ，k++，利用式(3)来调整参数 $step$ 和 $visual$，执行步骤 f)；否则，转 l)。

l)输出公告板中 X_{best} 及其函数值 $F(X_{best})$。

2　实验

HUST-CAID 系统是由哈尔滨理工大学计算机应用技术研究所开发的，用于设计 CAD 模型。系统采用 Visual C++.6.0 来实现。将本文所提出的方法应用于 HUST-CAID 系统的几何约束求解模块，以验证本文所提出方法的有效性。利用 HUST-CAID 系统建立源模型，如图 1 所示。

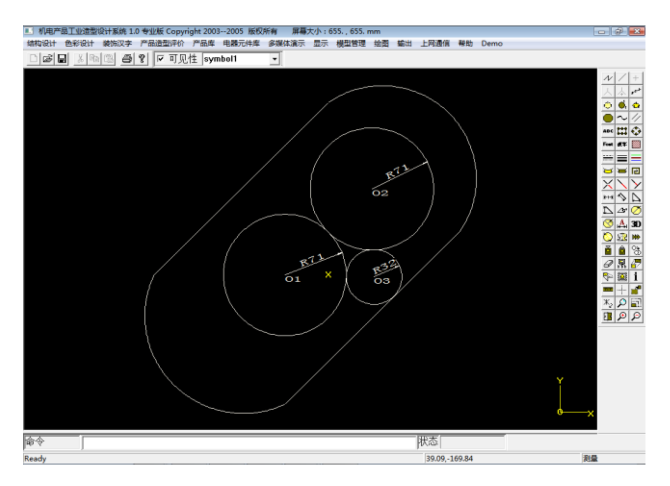


图 1　源模型

源模型为在两侧为圆弧的矩形板上开了 3 个圆孔。圆 1 的圆心为(-50, 0)，半径为 $50\sqrt{2}$；圆 2 的圆心为(50, 100)，半径为 $50\sqrt{2}$；圆 3 的圆心为(52.5, -2.5)，半径为 $45\sqrt{2}/2$；矩形板的一条边为直线 L，其方程如式(10)所示。在该几何模型中，要求满足 3 个约束条件。圆 1 和圆 3 相切；圆 2 和圆 3 相切；圆 3 与直线 L 相切。

$$x-y-100=0 \tag{10}$$

当圆 1 和圆 2 的半径都变为 100 时，将破坏模型的几何约束关系。为了保持以上的几何约束关系，圆 3 的圆心(x_1, x_2)和半径 x_3 将要发生改变。根据以上几何约束关系，可以得到一个非线性方程组，如式(11)所示。

$$\begin{cases}\sqrt{(x_1+50)^2+x_2^2}-x_3-100=0\\ \sqrt{(x_1-50)^2+(x_2-100)^2}-x_3-100=0\\ \dfrac{-x_1+x_2+100}{\sqrt{2}}-x_3=0\end{cases} \tag{11}$$

根据式(2)来构造目标函数 $F(X)$，如式(12)所示。

$$F(X)=\left(\sqrt{(x_1+50)^2+x_2^2}-x_3-100\right)^2+\left(\frac{-x_1+x_2+100}{\sqrt{2}}-x_3\right)^2+\left(\sqrt{(x_1-50)^2+(x_2-50)^2}-x_3-100\right)^2 \tag{12}$$

对式(11)所示的代数方程组进行人工求解，可以得到：x_1 的精确解为 $\left(-225+250\sqrt{2}\right)/2$，$x_2$ 的精确解为 $\left(325-250\sqrt{2}\right)/2$，$x_3$ 的精确解为 $\left(-500+375\sqrt{2}\right)/2$。

为了验证本文所提出方法的有效性，共进行了 3 组对比实验。在实验 1 中，利用鱼群算法来搜索使目标函数 $F(X)$最小化的 X 的值。在各次迭代过程中，所获取的全局最优解如表 1 所示。

表 1　鱼群算法各次迭代的全局最优解和函数值

迭代次数	x_1	x_2	x_3	$F(X)$
1	44.918049677	42.532715971	42.573079461	9391.492079168
5	50.163691530	29.824729964	15.292300532	3835.792710413
11	59.600146527	-5.751780193	16.612155450	217.983643780
33	61.014631057	-11.853944690	15.953236851	41.659222416
59	63.025702789	-16.367897430	16.777285954	11.586854095
1483	62.022162797	-14.441899056	15.046597387	6.946141181
1582	63.445447180	-13.941893489	14.173592186	3.625788803
1712	63.000566128	-14.710041460	14.588895029	2.508761004
2169	63.352987118	-14.976752801	14.812706610	1.363491782
2194	63.745531027	-14.529802641	15.262353774	0.368975245

从表 1 可以看出,鱼群算法在第 2194 次迭代中获得了最优解。x_1 的最优值为 63.745531027，x_2 的最优值为-14.529802641，x_3 的最优值为 15.262353774，$F(X)$的最优值为 0.368975245。

在实验 2 中，利用本文所提出的改进鱼群算法来搜索使目标函数 $F(X)$最小化的 X 的值。在各次迭代过程中，所获取的全局最优解如表 2 所示。

从表 2 可以看出,改进鱼群算法在第 784 次迭代中获得了最优解。x_1 的最优值为 64.440139669，x_2 的最优值为-14.447227389，x_3 的最优值为 15.362495618，$F(X)$的最优值为 0.188276500。

目前，粒子群算法被广泛地应用于函数优化问题。在实验 3 中，利用粒子群算法来搜索使目标函数 $F(X)$最小化的 X 的值[14]。在各次迭代过程中，所获取的全局最优解如表 3 所示。

表 2　改进鱼群算法各次迭代的全局最优解和函数值

迭代次数	x_1	x_2	x_3	$F(X)$
1	13.433212671	11.392035136	13.294932375	5823.022311359
3	24.351671842	6.993509501	15.071088575	3857.715236888
10	70.394742726	-2.461992450	12.819360556	168.071955844
13	69.404992310	-10.415457193	18.472989875	60.1095058804
16	66.947563325	-9.383974933	13.559048790	32.501231862
38	62.001200065	-13.623315915	16.175446820	16.025789995
145	67.468607697	-12.717839755	14.988403576	11.839974557
782	62.723831564	-16.227194083	15.606681150	5.2127209534
784	64.440139669	-14.447227389	15.362495618	0.188276500

表 3　粒子群算法各次迭代的全局最优解和函数值

迭代次数	x_1	x_2	x_3	$F(X)$
1	0.947493542	10.256312015	20.109044649	8231.060008148
5	6.540738582	7.614471435	19.437942504	6900.146019210
11	42.174404144	-19.904806137	21.201372146	756.318507301
15	55.333858489	-18.359180450	17.046781539	106.984610363
127	64.512969970	-13.938835144	16.805196762	8.339549484
189	63.161537170	-15.774456024	14.334611892	5.096380299
262	63.211776733	-14.562725067	14.204966545	3.534084232
287	63.471771240	-14.137941360	14.424797058	2.242321970
325	64.240600585	-14.031445503	14.378672599	1.779835471
1263	63.942073822	-14.404706001	15.385561943	0.311520001

从表 3 可以看出,粒子群算法在第 1263 次迭代中获得了最优解。x_1 的最优值为 63.942073822，x_2 的最优值为-14.404706001，

x_3的最优值为 15.385561943，$F(X)$的最优值为 0.311520001。

对比表 1 和 2 可以看出,实验 2 的 $F(X)$的最优值要小于实验 1 的 $F(X)$的最优值。因此，实验 2 所获取的全局最优解要优于实验 1 所获取的全局最优解。相对于鱼群算法而言，本文所提出的算法的全局最优解要更接近于人工求解结果。此外，实验 2 在第 784 次迭代中获得了最优解，其收敛速度要远远高于实验 1。其原因是：实验 2 使用混沌算法来改进鱼群算法，使其能快速地跳出局部最优，提高了搜索效率。对比表 2 和 3 可以看出,实验 2 的 $F(X)$的最优值要小于实验 3 的 $F(X)$的最优值。因此，实验 2 所获取的全局最优解要优于实验 3 所获取的全局最优解。本文所提出的方法的收敛速度要高于实验 3。

3　结束语

本文使用代数方程来表示几何约束关系。通过构造目标函数，将几何约束求解转换为函数优化问题。为了提高搜索求解效率，利用混沌算法来改进鱼群算法，以寻找目标函数的最优解。实验结果表明：相对于鱼群算法而言，本文所提出的方法无论在求解质量还是求解效率上都有所提高。

参考文献：

[1] 曹春红, 唐川, 赵大哲, 等. 量子行为粒子群优化算法在几何约束问题上的应用 [J]. 东北大学学报: 自然科学版, 2011, 32 (9): 1230-1232.

[2] 戴春来. 基于自由度分析的耦合几何约束求解 [J]. 计算机辅助设计与图形学学报, 2010, 22 (12): 2169-2176.

[3] 周友行, 张建勋, 董银松, 等. 基于约束问题求解的方向可选指数进制步长优化算法 [J]. 中南大学学报: 自然科学版, 2011, 42 (5): 1327-1331.

[4] 曹春红, 王鹏, 曹犁歌. 基于 D-tree 分解的欠约束与完备约束的几何约束求解 [J]. 东北大学学报: 自然科学版, 2014, 35 (5): 627-629.

[5] Thierry S E. A particle-spring approach to geometric constraints solving [C]// Proc of the 26th ACM Symposium on Applied Computing. 2011: 1100-1105.

[6] Mathis P, Schreck P. Coordinate-free geometry and decomposition in geometrical constraint solving [J]. CAD Computer Aided Design, 2014, 50 (1): 51-60.

[7] Gao Xiaoshan, Lin Qiang, Zhang Guifang. A C-tree decomposition algorithm for 2D and 3D geometric constraint solving [J]. CAD Computer Aided Design, 2006, 38 (1): 1-13.

[8] De Regt R, Van der Meiden H A, Hilderick A, et al. A workbench for geometric constraint solving [J]. Computer-Aided Design and Applications, 2008, 5 (1-4): 471-482.

[9] 郭希娟, 吴金垒. D-优化算法实现几何约束问题的重组 [J]. 小型微型计算机系统, 2011, 32 (9): 1891-1893.

[10] 曹春红, 唐川, 赵大哲, 等. 基于雁群启示的粒子群优化算法的几何约束求解 [J]. 小型微型计算机系统, 2011, 32 (11): 2299-2302.

[11] 曹春红, 易荣庆, 曹海龙, 等. 交互更新模式的量子遗传算法的几何约束求解 [J]. 中国图像图形学报, 2016, 21 (4): 499-509.

[12] 黄学良, 王蕾, 王旗华, 等. 三维几何约束系统的在线增量求解方法 [J]. 计算机辅助设计与图形学学报, 2017, 29 (8): 1571-1578.

[13] 李文辉, 孙明玉, 曹春红. 几何约束求解的扩展 C-树分解法 [J]. 吉林大学学报: 工学版, 2017, 47 (4): 1274-1279.

[14] 王书亭, 王战江. 粒子群优化算法求解非线性问题的应用研究 [J]. 华中科技大学学报: 自然科学版, 2005, 33 (12): 4-7.